
Constraint Handling Using Tournament Selection: Abductive Inference in Partly Deterministic Bayesian Network

Severino F. Galán

seve@dia.uned.es

Department of Artificial Intelligence, UNED, Madrid, 28040, Spain

Ole J. Mengshoel

omengshoel@riacs.edu

RIACS, NASA Ames Research Center, Moffett Field, CA, 94035, USA

Abstract

Constraints occur in many application areas of interest to evolutionary computation. The area considered here is Bayesian networks (BNs), which is a probability-based method for representing and reasoning with uncertain knowledge. This work deals with constraints in BNs and investigates how tournament selection can be adapted to better process such constraints in the context of abductive inference. Abductive inference in BNs consists of finding the most probable explanation given some evidence. Since exact abductive inference is \mathcal{NP} -hard, several approximate approaches to this inference task have been developed. One of them applies evolutionary techniques in order to find optimal or close-to-optimal explanations. A problem with the traditional evolutionary approach is this: As the number of constraints determined by the zeros in the conditional probability tables grows, performance deteriorates because the number of explanations whose probability is greater than zero decreases. To minimize this problem, this paper presents and analyzes a new evolutionary approach to abductive inference in BNs. By considering abductive inference as a constraint optimization problem, the novel approach improves performance dramatically when a BN's conditional probability tables contain a significant number of zeros. Experimental results are presented comparing the performances of the traditional evolutionary approach and the approach introduced in this work. The results show that the new approach significantly outperforms the traditional one.

Keywords

Constraint optimization problem, genetic algorithm, Bayesian network, approximate abductive inference, most probable explanation.

1 Introduction

Constraint handling by evolutionary algorithms has great importance, since many real-world problems are constrained. Among such problems are those that can be modeled by means of Bayesian networks (BNs) (Pearl, 1988; Jensen, 2001; Castillo et al., 1997). This work focuses on constraints that give rise to the appearance of zero values in the parametric part of a BN.

BNs (also known as *Bayesian belief networks*, *belief networks*, or *causal probabilistic networks*) are a type of probabilistic graphical model that became popular in the early nineties and has ever since been used for representing and reasoning with uncertain knowledge in a great variety of domains (Heckerman et al., 1995). They have been successfully applied in fields like estimation of distribution algorithms (Larrañaga and

Lozano, 2002), medical diagnosis (Heckerman et al., 1992; Heckerman and Nathwani, 1992), fault diagnosis and prediction in industry (Galán et al., 2007), computer vision (Rimey and Brown, 1994), user modeling (Horvitz et al., 1998), collaborative filtering (Pennock et al., 2000), and information retrieval (Fung and del Favero, 1995). Abductive inference and belief updating are the two main inference tasks that can be carried out in a BN. While belief updating calculates the posterior probability of each unobserved variable given some evidence (set of observed variables), abductive inference determines the assignment of values to the unobserved variables (also known as *explanation*) with maximum posterior probability given some evidence. Although methods exist that allow exact abductive inference to be performed in relatively complex networks, the problem has been proven to be \mathcal{NP} -hard (Shimony, 1994). As an example, in the case of clustering algorithms (Dawid, 1992; Seroussi and Goldmard, 1994; Nilsson, 1998) applied to highly connected networks, the cliques in the junction tree become very large, thus making abductive inference slow. For complex networks to which exact methods cannot be applied, approximate algorithms based on evolutionary computing (Lin et al., 1990; Rojas-Guzmán and Kramer, 1993, 1996; Gelsema, 1995; Welch, 1996; Mengshoel and Wilkins, 1998; Zhong and Santos Jr, 2000; de Campos et al., 2002) are an alternative that allows optimal or close-to-optimal solutions to be reached. As will be seen in this article, it turns out that great care is needed when using an evolutionary approach to performing inference in BNs with many zeros.

Extending previous research on constraint handling using tournament selection (Deb, 2000; Coello and Montes, 2002), this paper presents and analyzes a new approach to approximate abductive inference in BNs. The underlying idea is to consider this inference task as a constraint optimization problem (COP) such that an explanation is considered not to meet all of the constraints when its probability is equal to zero. In this way, the infeasible region is formed by explanations with zero probability. By establishing the goodness of an explanation from the number of constraints it violates (the more constraints violated, the worse the explanation is), the feasible region of explanations can be reached in less time than in the case of the traditional approach, where no constraints are explicitly considered. This leads to computational time savings, since the search for the optimal solution is efficiently restricted to the feasible region.

In this article, an analytical study is conducted to identify the differences between the traditional evolutionary approach to abductive inference in BNs and the new approach. Experimental tests are also reported comparing the performance of both approaches. The existing evolutionary approximate methods for abductive inference have the important disadvantage that their performance worsens as the number of zeros present in the BN conditional probability tables grows. In such cases, both efficiency and efficacy of the inference process deteriorate as a consequence of the space of explanations being dominated by those whose probability is equal to zero. The new approach overcomes this problem and leads to dramatic performance improvements when a significant number of zeros is contained in the conditional probability tables of a BN.

The rest of this paper is organized as follows. Section 2 reviews BNs and inference through evolutionary methods in BNs. Section 3 introduces the new evolutionary approach to approximate abductive inference. Section 4 classifies and analyzes the new approach within the context of constraint handling by evolutionary algorithms. Section 5 presents experimental results obtained for the comparison of the traditional and the new approaches. Section 6 contains a discussion derived from the present work. Finally, Section 7 concludes the paper and suggests future research directions.

2 Preliminaries

Every node in a Bayesian network (BN) (Pearl, 1988; Jensen, 2001; Castillo et al., 1997) represents a random variable. Nodes are connected by arcs that establish probabilistic dependence relations between variables. A BN is formed by an acyclic directed graph, along with a set of conditional probability distributions, usually represented as conditional probability tables (CPTs), one per node. A CPT contains the probability of a node, given any possible configuration of values for its parents. For root nodes, only their prior probabilities are needed. Given a BN with nodes $V = \{V_1, \dots, V_n\}$, the joint probability over the random variables in the network is expressed as

$$P(v_1, \dots, v_n) = \prod_{i=1}^n P(v_i \mid pa(v_i)), \quad (1)$$

where $pa(v_i)$ stands for a configuration of the set of parents for variable V_i . In this work, only discrete variables are considered. Belief updating amounts to fixing the values of the observed variables and calculating the posterior probability of each of the unobserved ones. The present work deals with another important type of inference process: *abductive inference*, also known as *belief revision*.

In the context of abductive inference, given some evidence or set of observations, $\mathbf{e} = \{E_1 = e_1, \dots, E_m = e_m\}$ with $E_i \in \{V_1, \dots, V_n\} \forall i$, an *explanation* is defined as a configuration of values for the BN variables, $\mathbf{v} = \{V_1 = v_1, \dots, V_n = v_n\}$, such that $\mathbf{e} \subset \mathbf{v}$. Among the set of possible explanations, abductive inference calculates the one with maximum posterior probability:

$$\mathbf{v}^* = \arg \max_{\mathbf{v}} P(\mathbf{v} \mid \mathbf{e}).$$

Usually, \mathbf{v}^* is referred to as *most probable explanation* (or MPE). A possible generalization of abductive inference can be obtained by considering the k most probable explanations. Without loss of generality, only the most probable explanation will be considered throughout this paper.

Since abductive inference in BNs has been proven to be an \mathcal{NP} -hard problem (Shimony, 1994), approximate methods were developed that allow optimal or close-to-optimal solutions to be obtained for complex networks. The reader is referred to (Gámez, 2004) for a complete review of exact and approximate methods for abductive inference in BNs. Although approximate abductive inference has been proven to be \mathcal{NP} -hard too (Abdelbar and Hedetniemi, 1998), the set of solvable problems is larger.

This work belongs to the type of approximate methods that use evolutionary computing to perform abductive inference in BNs. The representation of a candidate solution is frequently used to characterize the different dialects of evolutionary computation: evolutionary programming, genetic algorithms, evolution strategies, and genetic programming. This work employs genetic algorithms (GAs) (Holland, 1985; Goldberg, 1989), in which the **representation** of each candidate solution (or individual) uses a string whose elements are defined over a finite alphabet. In the context of abductive inference in BNs, each position (or gene) in the string corresponds to one variable in the BN. Each gene can take on a number of discrete values (or alleles) which correspond to the domain values of its associated variable. This simple representation of every explanation as a string is the most widely used for this problem, and the one selected for the present work.

The usual and simplest way to obtain an **initial population** in GAs is to initialize each gene independently and uniformly at random. This is the option selected in

this work for illustrative purposes: If no informed method is used to generate the initial individuals, the benefits of the novel approach can be more accurately evaluated. Therefore, the value for an unobserved variable is generated uniformly at random, while each evidence variable is assigned its observed value.

The definition of the **fitness** of an individual differentiates our approach from the traditional one. Typically, the fitness of an explanation is computed using Equation 1, i.e., it results from multiplying a set of factors, each taken from a different CPT after instantiating the BN variables to the corresponding explanation values. This is justified by the following result: $P(\mathbf{v} \mid \mathbf{e}) \propto P(\mathbf{v}, \mathbf{e})$. Since Druzdzel (1994) demonstrated that there are theoretical reasons to expect that a small number of complete instantiations of variables will account for most of the total probability mass, in a network with a substantial number of zeros in the CPTs, it could be expected that most of the genetic search would take place in a plateau formed by explanations of probability zero. This fact represents an important disadvantage for abductive inference to the point that it could be the case that no valid explanation could be reached in a reasonable time. Section 3 explains in detail the way this problem can be avoided by considering abductive inference in BNs as a constraint optimization problem (COP) and redefining the fitness function accordingly.

Selection and **variation** operators can be carried out both in traditional abductive inference through GAs, in general, and in the novel approach, in particular, by using the standard methods available in the literature. The reader is referred to (Bäck et al., 2000a,b) for a complete review of such methods.

The usual methods of handling **constraints** in evolutionary computation (Michalewicz and Schoenauer, 1996; Eiben, 2001; Coello, 2002) are based on the following techniques: penalty functions, repair functions, restriction of the search to the feasible region, and decoder functions.

The penalty function approach involves transforming constraints into optimization objectives, so that the constraints disappear and the problem is reduced to optimizing a modified fitness function. Usually, constraint violations are penalized through a function that reduces the fitness of individuals in proportion to the number of violated constraints. If a maximization problem is assumed, the objective function is mapped as follows given an individual s :

$$f'(s) = f_{obj}(s) - f_{pen}(s). \quad (2)$$

Function $f_{pen}(s)$ is called *penalty function* and is normally defined as a weighted sum:

$$f_{pen}(s) = \sum_{i=1}^p \omega_i \cdot d_i(s)^k, \quad (3)$$

where p is the number of constraints, ω_i is the weight associated with i -th constraint, $d_i(s)$ is a distance metric which measures the cost of repairing a violation of constraint i , and k is a constant often taking values 1 or 2.

Constraints can also be handled by means of repair functions or restriction of search. On the one hand, a mechanism can be used that repairs an individual violating constraints and produces another individual in which all constraints hold. On the other hand, specific representation, initialization, recombination, and mutation options can be adopted such that an individual is always guaranteed to meet the constraints.

Finally, an alternative technique for handling constraints is the use of a *decoder function* such that genotype is mapped to phenotype in a way that ensures that an in-

dividual is always mapped to a feasible solution. Different genotypes may be mapped onto the same phenotype by following this technique.

While different constraint handling techniques have been discussed in the literature, the remainder of this paper focuses on a hybrid approach that combines feasibility considerations with the use of penalty functions: constraint handling using tournament selection (Deb, 2000; Coello and Montes, 2002; Jiménez and Verdegay, 1999).

3 Constraints in Bayesian Networks

Abductive inference in BNs can be considered as a COP formed by tuple $\langle V, D, C, f_{obj} \rangle$. Given a network, the elements of the tuple can be defined as follows:

- $V = \{V_1, \dots, V_n\}$ is the set of n random variables in the BN.
- $\{D_1, \dots, D_n\}$ is the set of domains for the variables such that V_i is defined over D_i . A COP is defined in terms of variables V taking values within search space $D = \{D_1 \times \dots \times D_n\}$. A complete assignment of values to variables, denoted as Θ , is called *valuation*. Θ is a vector in D such that a value from D_i is assigned to each variable V_i , $\forall i \in \{1, \dots, n\}$. In this work, the set of mutually exclusive and exhaustive values within D_i are assumed to be discrete. Given some evidence $\mathbf{e} = \{E_1 = e_1, \dots, E_m = e_m\}$ with $m < n$ and $E_i \in V \forall i$, only valuations with variables $\{E_1, \dots, E_m\}$ instantiated to their corresponding values in \mathbf{e} will be considered.
- $C = \{C_1, \dots, C_n\}$ is a set of n constraints, where $C_i : D \rightarrow \{\text{true}, \text{false}\}$ are predicates over one or more variables in V , denoted as $C_i(V)$. A valuation Θ for which constraints hold, i.e., $C(\Theta) = \text{true}$, is called a *solution*. Each variable V_i , with $i \in \{1, \dots, n\}$, defines constraint C_i for any valuation (or explanation in BN terminology) $\{v_1, \dots, v_i, \dots, v_n\}$ as follows:

$$C_i(\{v_1, \dots, v_i, \dots, v_n\}) = \begin{cases} \text{false} & \text{if } P(v_i \mid pa(v_i)) = 0 \\ \text{true} & \text{otherwise} \end{cases}$$

where $pa(v_i)$ represents a configuration of the set of parents for variable V_i in explanation $\{v_1, \dots, v_i, \dots, v_n\}$. Note that $P(v_i \mid pa(v_i))$ can be obtained from the CPTs in the network. In other words, each variable V_i defines a constraint which is violated by explanation $\{v_1, \dots, v_i, \dots, v_n\}$ when entry $(v_i, pa(v_i))$ is equal to 0 in the CPT associated with V_i .

- $f_{obj} : D \rightarrow (0, 1]$ is the objective function assigning a numerical quality value to a solution. It is defined from Equation 1 for every explanation $\{v_1, \dots, v_i, \dots, v_n\}$ satisfying all of the constraints.

The goal is to find an optimal solution, i.e., a valuation such that the constraints are satisfied and the objective function is maximized.

3.1 The New Approach

The main part of the new evolutionary approach to abductive inference in BNs is its formalization of the fitness function. In order to introduce this formalization, two definitions are necessary:

Definition 1 Given a BN and an explanation $\mathbf{v} = \{v_1, \dots, v_n\}$, $z(\mathbf{v})$ stands for the number of constraints violated by \mathbf{v} .

Definition 2 Given a BN and an explanation $\mathbf{v} = \{v_1, \dots, v_n\}$, $f(\mathbf{v})$ denotes the result obtained by applying Equation 1 to \mathbf{v} after first discarding all factors equal to 0; therefore, $0 < f(\mathbf{v}) \leq 1 \forall \mathbf{v}$. Function $f(\mathbf{v})$ will be called “positive fitness”, since it is the traditional fitness restricted to include only positive conditional probabilities of an explanation.

The novel fitness function is constructed by taking into account both the number of constraints violated by an explanation and its positive fitness.

Definition 3 The dual fitness of an explanation \mathbf{v} is a pair $\langle z(\mathbf{v}), f(\mathbf{v}) \rangle$, following the previous definitions for $z(\mathbf{v})$ and $f(\mathbf{v})$.

Definition 4 An explanation \mathbf{v} is feasible if $P(\mathbf{v}) > 0$, whereas it is infeasible if $P(\mathbf{v}) = 0$.

The conditions under which an explanation is fitter than another are established in the following way (see (Deb, 2000) for a similar criterion):

Definition 5 Given two explanations \mathbf{v}_1 and \mathbf{v}_2 , the following cases can be considered in order to establish which explanation is fitter (and therefore the winner):

(i) If \mathbf{v}_1 and \mathbf{v}_2 are both feasible, i.e., they are solutions in which all of the constraints hold,

$$\text{winner} := \arg \max_{\mathbf{v} \in \{\mathbf{v}_1, \mathbf{v}_2\}} f(\mathbf{v}).$$

(ii) If one of \mathbf{v}_1 and \mathbf{v}_2 is feasible while the other is infeasible, the winner is the feasible one.

(iii) If \mathbf{v}_1 and \mathbf{v}_2 are both infeasible such that $z(\mathbf{v}_1) \neq z(\mathbf{v}_2)$,

$$\text{winner} := \arg \min_{\mathbf{v} \in \{\mathbf{v}_1, \mathbf{v}_2\}} z(\mathbf{v}).$$

(iv) If \mathbf{v}_1 and \mathbf{v}_2 are both infeasible such that $z(\mathbf{v}_1) = z(\mathbf{v}_2)$,

$$\text{winner} := \arg \max_{\mathbf{v} \in \{\mathbf{v}_1, \mathbf{v}_2\}} f(\mathbf{v}).$$

An explanation \mathbf{v}_1 being fitter than another one \mathbf{v}_2 will be denoted as $\mathbf{v}_1 \succ \mathbf{v}_2$. If there is a tie after cases (i)-(iv) have been applied, then a winner is picked uniformly at random.

Definition 3 gives rise to a type of fitness function that can be used with selection methods in which only an ordering by quality needs to be established for individuals. This is the case for *tournament selection* or *ranking selection*. However, selection methods based on assigning a numerical value to each individual’s fitness, like in *fitness proportional selection*, are not applicable when using this new approach.

Since the fitness function in Definition 3 has two parts, $z(\mathbf{v})$ and $f(\mathbf{v})$, such that $z(\mathbf{v})$ should be minimized and $f(\mathbf{v})$ should be maximized, the new approach involves the use of a dual-objective fitness function $\langle z(\mathbf{v}), f(\mathbf{v}) \rangle$. However, in terms of selection a single-objective GA is being used and not a multi-objective GA (Deb, 2001; Coello et al., 2002), since an ordering by quality is always possible for any pair of individuals in the new approach. The following example illustrates this point.

Example 1 Let \mathbf{v}_1 and \mathbf{v}_2 be two explanations for a BN, with fitness values $\langle z(\mathbf{v}_1) = 3, f(\mathbf{v}_1) = 0.9 \rangle$ and $\langle z(\mathbf{v}_2) = 1, f(\mathbf{v}_2) = 0.7 \rangle$, respectively. Since $z(\mathbf{v}_1) > z(\mathbf{v}_2)$ and $f(\mathbf{v}_1) > f(\mathbf{v}_2)$, \mathbf{v}_1 does not dominate \mathbf{v}_2 and \mathbf{v}_2 does not dominate \mathbf{v}_1 . However, under the approach introduced in this paper, we conclude using Definition 5 that $\mathbf{v}_2 \succ \mathbf{v}_1$, since $z(\mathbf{v}_2) < z(\mathbf{v}_1)$.

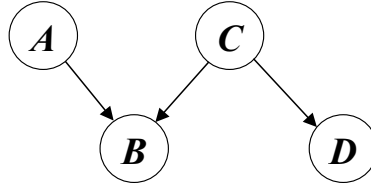


Figure 1: A BN with four nodes.

As will be shown in Section 5, an evolutionary algorithm based on a fitness function as established in Definition 3 and selection operating under Definition 5, often allows explanations satisfying all of the constraints to be more efficiently reached than in the case of an evolutionary algorithm where constraints are not incorporated into the fitness function. In this latter case, if the CPTs contain enough zeros, much of the genetic search would take place in a plateau formed by infeasible explanations. For networks whose CPTs contain no zeros, the novel fitness function and the traditional one become equivalent, since every explanation \mathbf{v} will satisfy the constraints: $\forall \mathbf{v} \ z(\mathbf{v}) = 0$.

Stochastic local search algorithms using similar ideas, such as GSAT and WalkSAT (Selman and Kautz, 1993; Hoos and Stützle, 2005), have been investigated for the satisfiability problem. These and similar algorithms minimize the number of unsatisfied clauses in propositional formulas. This corresponds to our minimization of the number of violated clauses $z(v)$. What is different in our formulation here is that we decompose fitness into two parts $z(v)$ and $f(v)$, and declare a winner as presented in Definition 5. Algorithms for satisfiability, including GSAT and WalkSAT, do not need to consider $f(v)$ since this part deals with probabilities, which is beyond the scope of propositional logic.

3.2 An Example

Consider the network depicted in Figure 1, formed by four binary variables whose domain is $\{0, 1\}$. The CPTs are defined by the following probabilities:

$$P(A = 0) = 0$$

$$P(B = 0 \mid A = 0, C = 0) = 0.4$$

$$P(B = 0 \mid A = 0, C = 1) = 0.7$$

$$P(B = 0 \mid A = 1, C = 0) = 0$$

$$P(B = 0 \mid A = 1, C = 1) = 0.6$$

$$P(C = 0) = 0.25$$

$$P(D = 0 \mid C = 0) = 0$$

$$P(D = 0 \mid C = 1) = 0.1.$$

Suppose that there is no evidence. The following probabilities are involved for explanation $\mathbf{v}_1 = (A = 0, B = 1, C = 0, D = 0)$:

$$P(A = 0) = 0$$

$$P(B = 1 \mid A = 0, C = 0) = 0.6$$

$$P(C = 0) = 0.25$$

$$P(D = 0 \mid C = 0) = 0.$$

Consequently, the traditional approach would assign to \mathbf{v}_1 the fitness value: $0 \cdot 0.6 \cdot 0.25 \cdot 0 = 0$. However, the approach presented in this paper gives \mathbf{v}_1 the following values:

$$\langle z(\mathbf{v}_1) = 2, f(\mathbf{v}_1) = 0.6 \cdot 0.25 = 0.15 \rangle.$$

Given a different explanation $\mathbf{v}_2 = (A = 1, B = 1, C = 0, D = 0)$,

$$\begin{aligned} P(A = 1) &= 1 \\ P(B = 1 \mid A = 1, C = 0) &= 1 \\ P(C = 0) &= 0.25 \\ P(D = 0 \mid C = 0) &= 0. \end{aligned}$$

The fitness of \mathbf{v}_2 in the traditional approach would be equal to: $1 \cdot 1 \cdot 0.25 \cdot 0 = 0$. The new approach would give \mathbf{v}_2 the values:

$$\langle z(\mathbf{v}_2) = 1, f(\mathbf{v}_2) = 1 \cdot 1 \cdot 0.25 = 0.25 \rangle.$$

Note that while both explanations have the same fitness under the traditional approach, \mathbf{v}_2 is fitter than \mathbf{v}_1 under the new approach.

3.3 Control of Constraints in Bayesian Networks

The number of zeros present in the randomly-generated CPTs of a BN can be controlled through the following algorithm.

Algorithm 1 *Generation of random CPTs for a BN such that the number of zeros included in its CPTs can be controlled:*

Input:

A BN

ρ *parameter controlling the number of zeros in the CPTs: $\rho \in [0, 1]$*

Output:

A BN with random CPTs and controlled number of zeros in them

1. *For each variable V in the network:*

(a) *If V has parents, for each possible configuration of V 's parents, $pa(v)$, generate the CPT entries, $P(v^i \mid pa(v)) \forall i \in \{1, \dots, |V|\}$, in the following way:*

i. For $i = 1$ to $|V|$:

if $i < |V|$

if $\text{rnd}(0, 1) < \rho$

$P(v^i \mid pa(v)) = 0$

else

$P(v^i \mid pa(v)) = \text{rnd} \left(0, 1 - \sum_{j=1}^{i-1} P(v^j \mid pa(v)) \right)$

else

$P(v^i \mid pa(v)) = 1 - \sum_{j=1}^{|V|-1} P(v^j \mid pa(v))$

(b) *If V has no parents, generate the CPT entries, $P(v^i) \forall i \in \{1, \dots, |V|\}$, by substituting in step 1.(a).i $P(v^i \mid pa(v))$ and $P(v^j \mid pa(v))$ by $P(v^i)$ and $P(v^j)$, respectively.*

where

$|V|$ is the domain size of variable V ,
 v^i is the i -th value of the domain for variable V , and
 $\text{rnd}(r_1, r_2)$ is a random number between r_1 and r_2 .

This algorithm is used in the present work to generate partly deterministic CPTs in BNs (see (Mengshoel et al., 2006) for a study of algorithms for generating random BNs of varying difficulty levels). The degree of determinism or, in other words, the expected percentage of zeros in CPTs, is controlled by means of the input parameter ρ . The greater the value of ρ , the larger the number of CPT entries are, on average, set to zero. There is a difference between Algorithm 1 and related research in that Algorithm 1 only sets a BN's CPT entries, while other algorithms also create (given certain constraints) the structure of the BN (Mengshoel et al., 2006).

From the point of view of constraint handling in GAs, the advantage of this algorithm is that it allows us to systematically control the size of the feasible region versus the infeasible region by changing the value of just one parameter ρ . As ρ increases, the size of the infeasible region increases (and, therefore, the size of the feasible region decreases).

Theorem 1 For BNs whose parameters are generated by Algorithm 1, let v be a random state of node V . The probability $P_{0,V}$ that $P(v \mid \text{pa}(v)) = 0$ is $\frac{|V|-1}{|V|} \cdot \rho$, where $|V|$ is the domain size for V .

Proof. In Algorithm 1, values v^i ($i \in \{1, \dots, |V| - 1\}$) are assigned $P(v^i \mid \text{pa}(v)) = 0$ with probability equal to ρ . Value $v^{|V|}$ corresponds to last state for variable V , which is always used for CPT normalization and, consequently, assigned a conditional probability $P(v^{|V|} \mid \text{pa}(v)) > 0$. Therefore, a correction factor $\frac{|V|-1}{|V|}$ has to be applied to ρ in $P_{0,V}$. ■

One needs to be careful when discussing what makes BNs hard or easy; several issues are intertwined. There is the issue of the number of solutions. There is also the issue of BN structure (or topology), which again has impact on treewidth (or optimal maximal clique size in a clique tree) (Mengshoel et al., 2006). There are also the issues of evidence and the values of conditional probabilities. A comprehensive discussion of what makes BNs hard or easy is beyond the scope of this paper. Our main focus in this article is on determinism, and in particular how determinism is handled in different variants of tournament selection.

4 Constraint Handling Using Tournament Selection

The novel approach to abductive inference in BNs takes ideas from the penalty-function technique in order to handle constraints. By substituting $\omega_i = 1 \forall i, k = 1$, and

$$d_i(s) = \begin{cases} 1 & \text{if constraint } i \text{ is violated} \\ 0 & \text{if constraint } i \text{ is held} \end{cases}$$

in Equation 3, a penalty function $f_{\text{pen}}(\mathbf{v}) = z(\mathbf{v})$ is created that measures the number of constraints violated by explanation \mathbf{v} . However, instead of adding the penalty function to the objective function as in Equation 2, the penalty component $z(\mathbf{v})$ is explicitly stored together with the positive fitness $f(\mathbf{v})$. Subsequently, $z(\mathbf{v})$ and $f(\mathbf{v})$ can be used in turn —first $z(\mathbf{v})$ and then $f(\mathbf{v})$ — to select a winning explanation in tournament selection or to rank explanations in ranking selection. Therefore, a hybrid method is developed that combines:

- (i) consideration of feasible ($P(\mathbf{v}_{\text{feasible}}) > 0$) and infeasible ($P(\mathbf{v}_{\text{infeasible}}) = 0$) explanations, and
- (ii) a penalty function $z(\mathbf{v})$ which is only used when both explanations are infeasible.

The approach introduced in this work has some similarities with methods that use a penalty-function approach but do not require any penalty parameter ω_i . Such methods (Deb, 2000; Jiménez and Verdegay, 1999) use tournament selection based on the following rules:

- (i) When two feasible individuals are compared, the one with better objective function value is chosen.
- (ii) When one feasible individual is compared with an infeasible one, the feasible individual is chosen.
- (iii) When two infeasible individuals are compared, the one with smaller constraint violation is chosen.

Another method that uses tournament selection in the context of constraint optimization problems can be found in (Coello and Montes, 2002). The underlying idea is to use concepts from multi-objective optimization to derive constraint-handling techniques. This method redefines the single-objective optimization function f_{obj} as a multi-objective optimization problem with $p + 1$ objectives, where p is the number of constraints; then any multi-objective optimization technique can be applied to the vector $(f_{obj}, f_1, \dots, f_p)$, where $\{f_1, \dots, f_p\}$ are the original constraints of the problem. A set of rules based on dominance relations between individuals and feasibility of individuals are applied in order to carry out the selection process.

The approach proposed in the present work is more similar to that of Deb (2000) (see also (Deb, 2001, Sections 7.5 and 4.2.3), where the method is called “constrained tournament method”), who uses single-objective tournament selection, than to that of Coello and Montes (2002), who use multi-objective tournament selection.

4.1 The Algorithm

The following *simple genetic algorithm* can be used to compare the traditional and the novel approaches to abductive inference in BNs.

Algorithm 2 *Simple genetic algorithm for abductive inference in BNs:*

Input:

FitnessFunction: *type of fitness used* (TraditionalFitness or NovelFitness)

m *population size*

n *number of nodes in the BN*

g *number of generations*

p_c *crossover rate*

p_m *mutation rate*

Output:

$\hat{\mathbf{v}}^*$ *An estimate of the MPE*

1. *Initialize the population with random candidate explanations.*

For $i = 1$ *to* m :

For $j = 1$ *to* n :

Assign to explanation \mathbf{v}_i a value for variable V_j chosen uniformly at random from V_j 's domain.

2. Do g times:

(a) Select parents.

For $i = 1$ to m :

Parent i is selected by tournament with tournament size equal to 2 and without replacement. Choose as winner the explanation with greatest joint probability, in the case of TraditionalFitness, or the fitter explanation from Definition 5, in the case of NovelFitness:

$$\text{Parent}_i := \text{Tournament}(\mathbf{v}_1, \mathbf{v}_2, \text{FitnessFunction}),$$

where \mathbf{v}_1 and \mathbf{v}_2 are two explanations selected uniformly at random from the current population.

(b) Recombine parents by single-point crossover with a given crossover rate p_c .

(c) Mutate offspring with a given mutation rate p_m for each gene or variable. Each new allele is selected uniformly at random among the rest of values for the corresponding variable.

3. Obtain from the final population: $\hat{\mathbf{v}}^* = \arg \max_{\mathbf{v}} P(\mathbf{v})$.

Algorithm 2 generates the initial population in a random way. Parent selection is performed by means of tournament selection. Single-point crossover is chosen as recombination operator. Mutation is carried out such that a new allele is chosen uniformly at random from the set of permissible ones. Finally, survivor selection is implemented as a generational model.

4.2 Analysis

Without loss of generality, in this section it will be assumed that no evidence is present.

Definition 6 Let $z(t)$ be a random variable denoting the number of constraints violated by an individual at generation t . Then $P(z(t) = i)$ is the probability that an individual at generation t violates i constraints.

The first step in Algorithm 2 implements random initialization of the population. The relative distribution of feasible versus infeasible individuals in the initial population is important for the comparative performance between the traditional and the novel approaches.

Theorem 2 Consider an arbitrary individual in the initial population for a BN generated by Algorithm 1. The probability that this individual is feasible is:

$$P(z(1) = 0) = \prod_{i=1}^n \left(1 - \frac{|V_i| - 1}{|V_i|} \cdot \rho \right).$$

Proof. Random generation of an individual consists of n random allele choices. A random allele choice for i -th gene gives rise to the random selection of a conditional probability from the CPT associated with the corresponding variable V_i . Let $P_{0,i}$ denote the probability that $P(v_i \mid pa(v_i)) = 0$. As n independent allele choices have to be made, the probability that n conditional probabilities greater than 0 are obtained in order to

generate a feasible individual is equal to $\prod_{i=1}^n (1 - P_{0,i})$. The theorem is immediate by taking into account that, from Theorem 1, $P_{0,i} = \frac{|V_i|-1}{|V_i|} \cdot \rho$ for BNs generated by Algorithm 1. ■

Consider large BNs with a great number of zeros in their CPTs. Once the initial population has been created, it is very likely that every individual violates at least one constraint. In the case of the traditional approach, this means that every individual in the initial population has fitness equal to zero and, consequently, a *random walk* would take place. Such random walk is avoided in the case of the novel approach, since different fitnesses would be assigned to the initial individuals. The mean number of generations that random walk would take for a given BN under the traditional approach is given by the following theorem. This random walk is approximated with random and independent generation of populations.

Theorem 3 *The mean generation at which one or more feasible individuals is generated is*

$$t_{\text{first}} = 1 / \left(1 - \left\{ 1 - \prod_{i=1}^n \left(1 - \frac{|V_i|-1}{|V_i|} \cdot \rho \right) \right\}^m \right).$$

Proof. From Theorem 2, the probability of a randomly generated individual being infeasible is

$$P(z(1) > 0) = 1 - \prod_{i=1}^n \left(1 - \frac{|V_i|-1}{|V_i|} \cdot \rho \right). \quad (4)$$

Define a successful outcome, with probability p_{success} , to be one in which there is one or more feasible individuals in the population. Define a failed outcome, with probability p_{failure} , to be one in which all m individuals in the population are infeasible. Clearly, $p_{\text{success}} + p_{\text{failure}} = 1$ and thus

$$\begin{aligned} p_{\text{success}} &= 1 - p_{\text{failure}} \\ &= 1 - P(z(1) > 0)^m. \end{aligned} \quad (5)$$

By substituting (4) into (5), the following is obtained:

$$p_{\text{success}} = 1 - \left\{ 1 - \prod_{i=1}^n \left(1 - \frac{|V_i|-1}{|V_i|} \cdot \rho \right) \right\}^m.$$

Assuming that populations are generated independently at random, the number of populations generated before success is a geometric random variable with probability p_{success} and mean $t_{\text{first}} = 1/p_{\text{success}}$, giving the desired result. ■

It should be noted that random walk can have a damaging effect on the traditional approach's performance. As an example, for $m = 100$, $P_{0,i} = \frac{|V_i|-1}{|V_i|} \cdot \rho = 0.2 \forall i \in \{1, \dots, n\}$, and $n = 50$, the random walk is expected to take place for 700 generations.

Once the random walk is over under the traditional approach, one individual appears which is fitter than the rest. If just selection were considered, this individual would take over the whole population after a number of generations. In the novel approach, the takeover process could be considered to begin in the first generation. Results estimating takeover times for different selection schemes can be found in (Goldberg and Deb, 1991, Section 5) and (Thierens and Goldberg, 1994, Section 3). Theorem 4 constitutes a particularization for the problem considered in this work of the results obtained by Goldberg and Deb (1991) for a takeover process under tournament selection.

Theorem 4 *For a simple genetic algorithm based on Algorithm 2, if just tournament selection is considered (recombination and mutation are not applied):*

$$P(z(t) > 0) = \left(1 - \prod_{i=1}^n (1 - P_{0,i})\right)^{\gamma^{t-1}},$$

where t specifies the current generation, $t = 1$ for the initial population, γ is the tournament size, and $P_{0,i}$ is the probability that a conditional probability equal to 0 is randomly selected from the CPT associated with i -th node in the BN.

Proof. The theorem will be proven by induction. It is immediate that $P(z(1) > 0) = 1 - P(z(1) = 0) = 1 - \prod_{i=1}^n (1 - P_{0,i})$, which proves the theorem for $t = 1$. Assuming that the theorem is true for generation t , an infeasible individual will be selected for generation $t+1$ if γ infeasible individuals from generation t are chosen for a tournament. Therefore,

$$\begin{aligned} P(z(t+1) > 0) &= \left(1 - \prod_{i=1}^n (1 - P_{0,i})\right)^{\gamma^{t-1}} \cdot \gamma \cdot \left(1 - \prod_{i=1}^n (1 - P_{0,i})\right)^{\gamma^{t-1}} \\ &= \left(1 - \prod_{i=1}^n (1 - P_{0,i})\right)^{\gamma^t}, \end{aligned}$$

which proves the theorem for generation $t + 1$. ■

Theorem 4 can be applied to both the traditional and the novel approaches. Therefore, this theorem does not show how different the latter works from the former. Instead of studying $P(z(t) > 0)$, Theorem 5 and Theorem 6 allow $P(z(t) = i) \forall i \in \{1, \dots, n\}$ to be calculated for the traditional and the novel approaches, respectively. Thus, a comparative study between the two approaches will be possible.

Theorem 5 (Constraints violated, traditional approach) *For a simple genetic algorithm based on Algorithm 2, if just tournament selection is considered under the traditional approach (recombination and mutation are not applied):*

$$P(z(t) = i) = \{1 - P(z(1) = 0)\}^{\gamma^{t-1}-1} \cdot P(z(1) = i) \quad \forall i \in \{1, \dots, n\},$$

where t specifies the current generation, $t = 1$ for the initial population, and γ is the tournament size.

Proof. A proof by induction will be given. It is immediate that the theorem holds for $t = 1$. An individual violating i constraints ($i \in \{1, \dots, n\}$) will be selected for generation $t+1$ if it is randomly selected from those of generation t chosen for a tournament, given that no feasible individual is chosen for the same tournament. Since the probability that no feasible individual is chosen at $t+1$ for a tournament is equal to $\{1 - P(z(t) = 0)\}^\gamma$,

$$\begin{aligned} P(z(t+1) = i) &= \{1 - P(z(t) = 0)\}^\gamma \cdot \frac{P(z(t) = i)}{\sum_{j=1}^n P(z(t) = j)} \\ &= \left(\sum_{j=1}^n P(z(t) = j)\right)^{\gamma-1} \cdot P(z(t) = i). \end{aligned}$$

Assuming that the theorem is true for generation t :

$$\begin{aligned} P(z(t+1) = i) &= \left(\{1 - P(z(1) = 0)\}^{\gamma^{t-1}-1} \cdot \sum_{j=1}^n P(z(1) = j) \right)^{\gamma-1} \cdot P(z(t) = i) \\ &= \left(\{1 - P(z(1) = 0)\}^{\gamma^{t-1}} \right)^{\gamma-1} \cdot P(z(t) = i). \end{aligned}$$

Applying again the inductive hypothesis for generation t :

$$\begin{aligned} P(z(t+1) = i) &= \{1 - P(z(1) = 0)\}^{\gamma^{t-1} \cdot (\gamma-1) + \gamma^{t-1} - 1} \cdot P(z(1) = i) \\ &= \{1 - P(z(1) = 0)\}^{\gamma^t - 1} \cdot P(z(1) = i) \quad \forall i \in \{1, \dots, n\}, \end{aligned}$$

which proves the theorem for generation $t + 1$. ■

Theorem 5 constitutes a generalization of the results obtained by Goldberg and Deb (1991) for the proportion of best-fitness individuals in a population subjected to tournament selection. Here, instead of two categories (optimal vs. non-optimal individuals), $n + 1$ categories are considered (one category for the feasible individuals and n categories for individuals violating i constraints, with $i \in \{1, \dots, n\}$). An individual from the category of feasible individuals wins over any other infeasible individual from an inferior category. For two infeasible individuals from different categories, under the traditional approach the winner is selected in a random way.

Theorem 6 (Constraints violated, novel approach) *For a simple genetic algorithm based on Algorithm 2, if just tournament selection is considered under the novel approach (recombination and mutation are not applied):*

$$P(z(t) = i) = \left(\sum_{j=i}^n P(z(1) = j) \right)^{\gamma^{t-1}} - \left(\sum_{j=i+1}^n P(z(1) = j) \right)^{\gamma^{t-1}} \quad \forall i \in \{1, \dots, n\},$$

where t specifies the current generation, $t = 1$ for the initial population, and γ is the tournament size.

Proof. The theorem will be proven by induction. It is immediate that the theorem is true for $t = 1$. An individual violating i constraints ($i \in \{1, \dots, n\}$) will be selected for generation $t + 1$ if it is chosen from generation t for a tournament and no individual violating fewer than i constraints is chosen for the same tournament. Equivalently, $P(z(t+1) = i)$ can be calculated by subtracting the probability that an individual violating more than i constraints is selected for generation $t + 1$ from the probability that γ individuals violating i or more than i constraints are selected from generation t for a tournament:

$$P(z(t+1) = i) = P(z(t) \geq i)^\gamma - P(z(t+1) \geq i+1) \quad (6)$$

$$= \left(\sum_{j=i}^n P(z(t) = j) \right)^\gamma - \left(\sum_{j=i+1}^n P(z(t+1) = j) \right). \quad (7)$$

From Equation 7 for $i = n$:

$$P(z(t+1) = n) = P(z(t) = n)^\gamma,$$

which combined with Equation 7 for $i = n - 1$ yields:

$$P(z(t+1) = n-1) = \left(\sum_{j=n-1}^n P(z(t) = j) \right)^\gamma - P(z(t) = n)^\gamma.$$

In general,

$$P(z(t+1) = i) = \left(\sum_{j=i}^n P(z(t) = j) \right)^\gamma - \left(\sum_{j=i+1}^n P(z(t) = j) \right)^\gamma \quad \forall i \in \{1, \dots, n\}. \quad (8)$$

Applying the inductive hypothesis for generation t in Equation 8:

$$\begin{aligned} P(z(t+1) = i) &= \left\{ \left(\sum_{j=i}^n P(z(1) = j) \right)^{\gamma^{t-1}} \right\}^\gamma - \left\{ \left(\sum_{j=i+1}^n P(z(1) = j) \right)^{\gamma^{t-1}} \right\}^\gamma \\ &= \left(\sum_{j=i}^n P(z(1) = j) \right)^{\gamma^t} - \left(\sum_{j=i+1}^n P(z(1) = j) \right)^{\gamma^t} \quad \forall i \in \{1, \dots, n\}, \end{aligned}$$

which proves the theorem for generation $t + 1$. ■

Similar to Theorem 5, Theorem 6 generalizes the work by Goldberg and Deb (1991). However, for two infeasible individuals from different categories (with different number of constraints violated), in the case of the novel approach the winner is selected based on the new fitness function introduced in Definition 3.

4.3 An Example

In order to compare the formula in Theorem 5 with the formula in Theorem 6, let the parameters involved be instantiated as follows: $n = 25$, $\gamma = 2$, and

$$P(z(1) = i) = \binom{n}{i} \cdot \left(1 - \frac{w-1}{w} \cdot \rho \right)^{n-i} \cdot \left(\frac{w-1}{w} \cdot \rho \right)^i \quad \forall i \in \{1, \dots, n\}, \quad (9)$$

where w is the mean domain size for the BN variables and is assigned value 2, and $\rho = 0.3$. Equation 9 corresponds to population initialization carried out uniformly at random. In this equation, the second factor multiplied by the third factor is the probability that an individual violating only i given constraints is created. Figure 2 depicts the graphs for $P(z(t) = i)$ ($i \in \{1, \dots, 8\}$) under the traditional and the novel approaches.

The graphs for the traditional approach show an exponential decay pattern for $P(z(t) = i) \forall i \in \{1, \dots, 8\}$. As far as the graphs for the novel approach are concerned, while a rapid exponential-decay pattern is observed for high i values (more rapid than in the case of the traditional approach), for low i values $P(z(t) = i)$ first grows to reach a maximum (the lower i , the higher the maximum) and then decreases to zero. Consequently, two main conclusions can be obtained from Figure 2:

- (i) The novel approach eliminates individuals with a high number of violated constraints more rapidly than the traditional approach does.

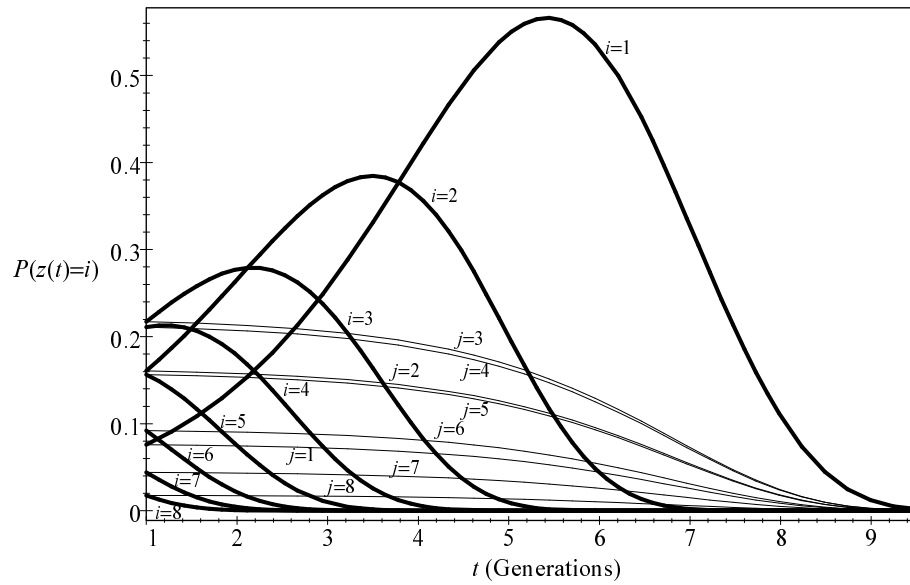


Figure 2: $P(z(t) = i)$ under the novel (thick lines, index i) and the traditional (thin lines, index j) approaches.

- (ii) Under the novel approach, individuals with few violated constraints rapidly take over the population.

Recombination and mutation make the performance of an evolutionary algorithm differ from a simple takeover process guided by selection only. The next section aims at empirically comparing the performances of two genetic algorithms which include recombination and mutation working under the traditional and the novel approaches, respectively.

5 Experiments

In order to compare the performance of the traditional approach with that of the novel approach, abductive inference was carried out on a set of BNs using both approaches. Without loss of generality, MPE computation with no evidence was performed.

The BNs used in the experiments were generated in two phases:

- Firstly, a number of BNs were selected from the literature devoted to abductive inference in BNs using evolutionary methods. The graphs and the domains of variables were maintained in the selected BNs. Table 1 shows the main characteristics of the four networks¹ finally selected.
- Secondly, random CPTs were generated for each of the four selected networks by applying Algorithm 1. BNs with random CPTs were generated independently of

¹Some BN repositories where Alarm network (Beinlich et al., 1989) can be obtained are:
<http://genie.sis.pitt.edu/networks.html>
<http://www.cs.huji.ac.il/labs/compbio/Repository/>
<http://www.cs.aux.dk/research/DSS/Misc/networks.html>

<i>BN Name</i>	<i>No. of Nodes</i>	<i>Search Space Size</i>	<i>Taken from</i>
Gelsema	Binary: 15	3.2768E+4	(Gelsema, 1995)
Rojas	Binary: 15 Ternary: 5	7.962624E+6	(Rojas-Guzmán and Kramer, 1996)
Welch	Binary: 32	4.294967295E+9	(Welch, 1996)
Alarm	Binary: 13 Ternary: 17 Quaternary: 7	1.7332899271409664E+16	(de Campos et al., 2002)

Table 1: Main characteristics of the four BNs selected from the literature. These networks are the basis for the 32 networks used in experiments.

each other for the four networks in Table 1 and for $\rho \in \{0.1, 0.2, \dots, 0.8\}^2$, giving a total of 32 BNs used for experimentation.

The simple genetic algorithm described in Section 4.1 was used in the experiments with $p_c = 1$ and $p_m = \frac{1}{n}$, where n is the number of nodes in the BN. Survivor selection was implemented as a generational model with elitism for the best individual.

Two types of experiments were conducted. In the first type, the results for both approaches after a fixed number of generations were compared. The second type studied population evolution generation by generation.

5.1 Results After a Fixed Number of Generations

Applying both the traditional and the novel approaches, the genetic algorithm was run one hundred times on each of 32 BNs generated randomly from the Gelsema, Rojas, Welch, and Alarm networks. Each run for a given BN with random CPTs was executed for $g = 50$ generations with the following population sizes: $m = 10$ for Gelsema networks, $m = 100$ for Rojas networks, $m = 150$ for Welch networks, and $m = 1000$ for Alarm networks. Different seeds for the random number generator were used for each run and each approach.

Tables 2-5 show the sample means of the best fitnesses reached in one hundred runs performed under both approaches for each of the randomly generated BNs. The percentage of runs reaching optimum in each BN experiment, alternatively known as *accuracy*, is also included in the tables.

Paired t -tests were performed on the best-fitness data contained in Tables 2-5 to determine whether the *null hypothesis* is consistent with the data, i.e., whether the results obtained for the traditional approach and those obtained for the novel approach come from the same distribution and, therefore, the differences are due to random effects. The results for paired t -tests, on pairs constituted by mean best fitness values under both approaches, are included in Table 6. Besides t value, probability P of the corresponding result assuming the null hypothesis is shown. The values for P in Table 6 can be considered as significant enough to discard the null hypothesis for the data obtained for $\rho \in \{0.1, 0.2, \dots, 0.8\}$, suggesting that the novel approach is better than the traditional approach.

²Value $\rho = 0$ was omitted because it represents the case in which, a priori, no zeros are introduced in the randomly generated CPTs and, as a consequence, both approaches perform the same. Values $\rho = 0.9$ and $\rho = 1$ were not considered because they constitute extreme cases, rarely appearing in real-world BNs.

ρ	<i>Best fitness</i>		<i>Accuracy (%)</i>	
	Traditional	Novel	Traditional	Novel
0.1	0.08574	0.08197	75	71
0.2	0.01159	0.01182	62	71
0.3	0.06665	0.07645	62	74
0.4	0.02005	0.02426	40	61
0.5	0.05881	0.08895	50	87
0.6	0.02577	0.16508	10	82
0.7	0.05044	0.27632	11	83
0.8	0.02756	0.36629	6	100

Table 2: Experimental results after a fixed number of generations for eight Gelsema networks over 100 runs.

ρ	<i>Best fitness</i>		<i>Accuracy (%)</i>	
	Traditional	Novel	Traditional	Novel
0.1	0.0021	0.00221	51	62
0.2	0.00186	0.00183	50	46
0.3	0.00106	0.00121	32	52
0.4	0.00989	0.05924	5	79
0.5	0.00583	0.02358	2	48
0.6	0.00906	0.11132	3	99
0.7	0.00142	0.21164	0	99
0.8	4.43639E-4	0.0982	0	97

Table 3: Experimental results after a fixed number of generations for eight Rojas networks over 100 runs.

ρ	<i>Best fitness</i>		<i>Accuracy (%)</i>	
	Traditional	Novel	Traditional	Novel
0.1	6.79454E-5	7.09712E-5	23	35
0.2	0.00152	0.00197	14	17
0.3	1.64007E-4	2.52125E-4	9	22
0.4	5.21534E-5	0.00427	0	15
0.5	1.0677E-5	0.00505	0	87
0.6	4.25574E-6	0.01275	0	33
0.7	0	0.04162	0	47
0.8	0	0.07485	0	92

Table 4: Experimental results after a fixed number of generations for eight Welch networks over 100 runs.

ρ	<i>Best fitness</i>		<i>Accuracy (%)</i>	
	Traditional	Novel	Traditional	Novel
0.1	1.03055E-7	2.08097E-7	0	0
0.2	1.44638E-10	8.67431E-8	0	0
0.3	1.29603E-11	5.89381E-7	0	0
0.4	2.06003E-15	7.9208E-6	0	0
0.5	2.82928E-12	1.06317E-5	0	0
0.6	5.20155E-20	1.25066E-4	0	0
0.7	0	2.16682E-4	0	0
0.8	0	0.01654	0	46

Table 5: Experimental results after a fixed number of generations for eight Alarm networks over 100 runs.

<i>Networks</i>	<i>t</i> (mean best fitness)	<i>P</i> (mean best fitness)
Gelsema	2.04	0.081
Rojas	2.28	0.057
Welch	1.82	0.112
Alarm	1.02	0.34

Table 6: Paired *t*-test results for the experiments measuring best fitness after a fixed number of generations under the traditional and the novel approaches.

5.2 Results Obtained Generation by Generation

Both approaches were executed one hundred times on the same type of networks described for the previous tests and using the same population sizes. However, $g = 1000$ generations were now studied by measuring the following two magnitudes generation by generation:

- *Best fitness*: For the best individual at generation t under each approach, its joint probability is reported, $P(\mathbf{v}_{best}(t))$.
- *Estimate of $P(z(t) > 0)$* : $P(z(t) > 0)$, defined in Section 4.2, is estimated by dividing the number of infeasible individuals at t by m . The analysis of the evolution of this estimate allows one to study the ability of the two approaches to find feasible solutions.

Figures 3-6 show the mean best fitness for the Gelsema, Rojas, Welch, and Alarm networks, respectively, for one hundred runs. Logarithmic scale was used to facilitate viewing the parts of the graphs with more variability. Figures 7-10 depict the mean $P(z(t) > 0)$ for the Gelsema, Rojas, Welch, and Alarm networks, respectively.

6 Discussion

6.1 Performance After a Fixed Number of Generations

Results in Section 5.1 for a fixed number of generations show that the performance of the traditional approach deteriorates compared to the novel approach as ρ grows from 0.1 to 0.8. In general, while the performance of the traditional approach worsens as ρ grows, that of the novel approach improves. Mean best fitness and mean accuracy

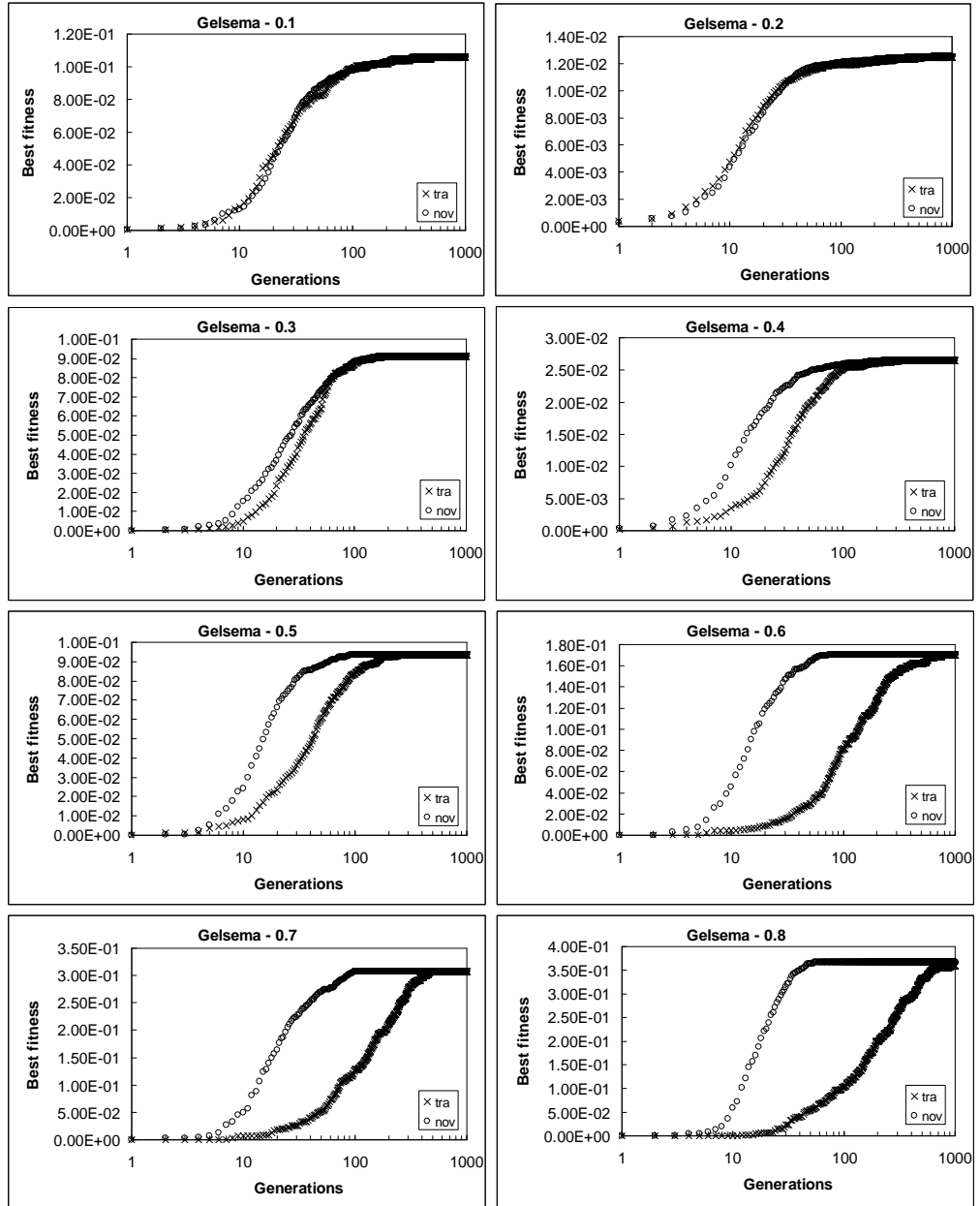


Figure 3: Best fitness results for eight Gelsema networks and $\rho \in \{0.1, \dots, 0.8\}$ under the traditional (tra) and the novel (nov) approaches.

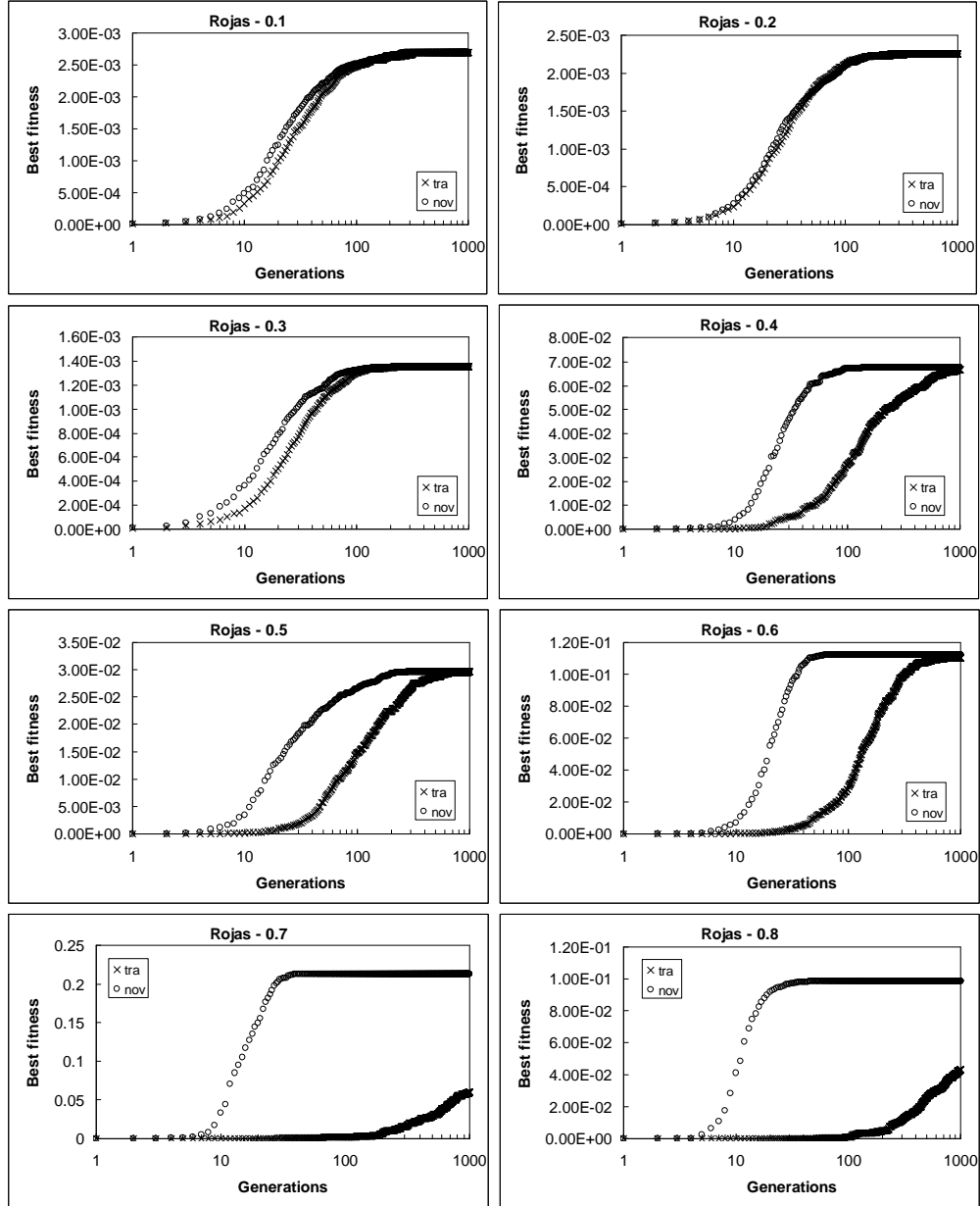


Figure 4: Best fitness results for eight Rojas networks and $\rho \in \{0.1, \dots, 0.8\}$ under the traditional (tra) and the novel (nov) approaches.

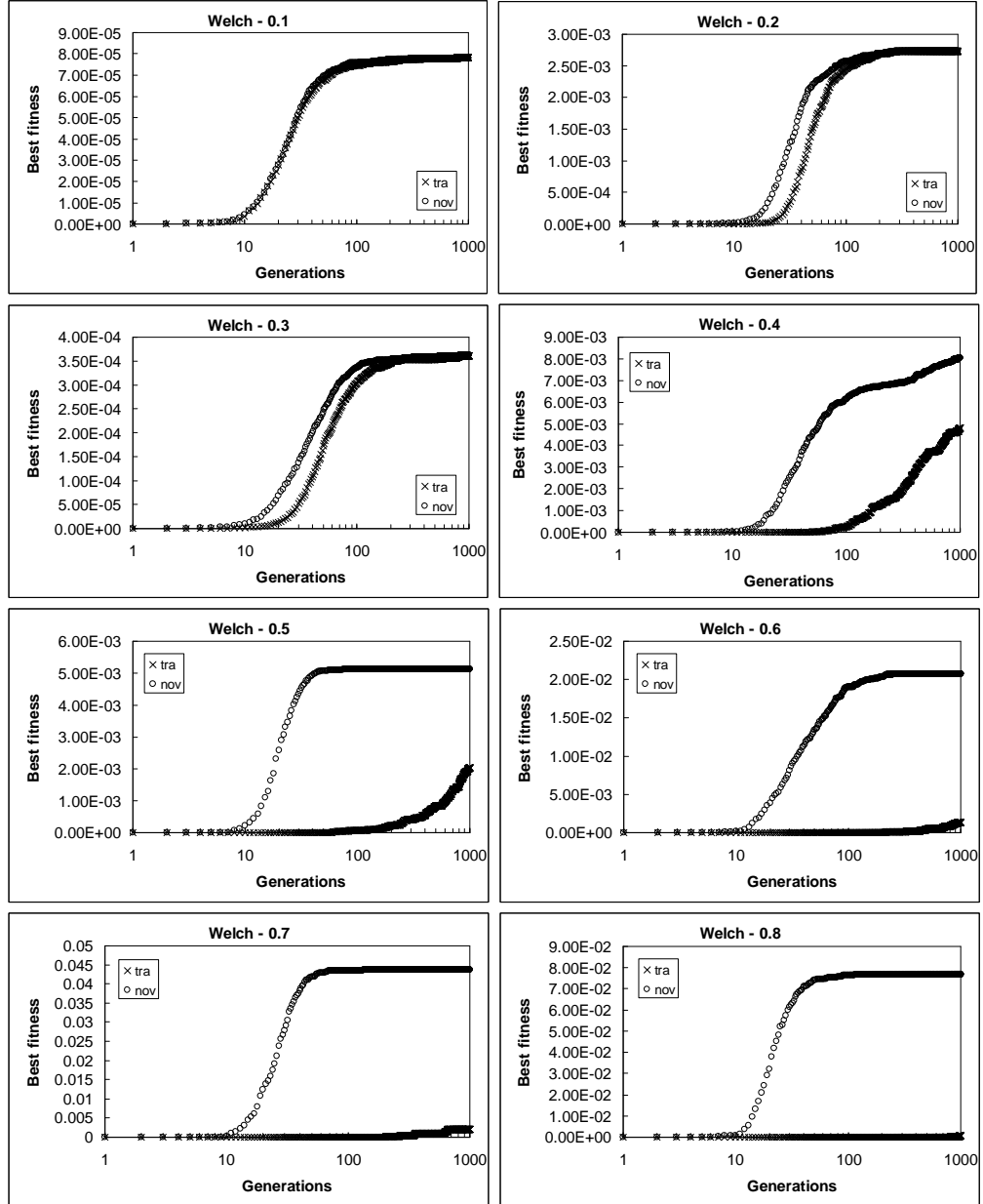


Figure 5: Best fitness results for eight Welch networks and $\rho \in \{0.1, \dots, 0.8\}$ under the traditional (tra) and the novel (nov) approaches.

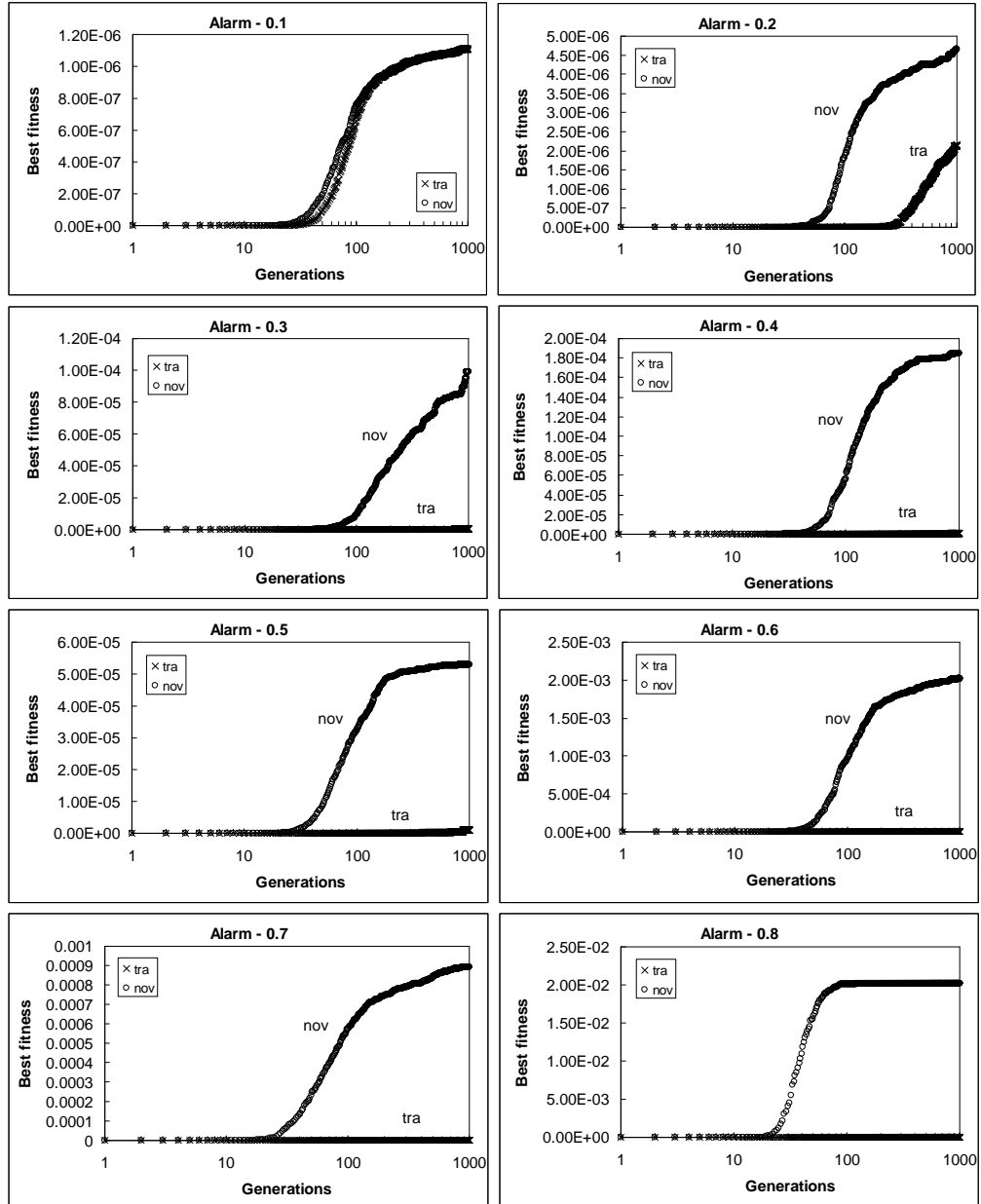


Figure 6: Best fitness results for eight Alarm networks and $\rho \in \{0.1, \dots, 0.8\}$ under the traditional (tra) and the novel (nov) approaches.

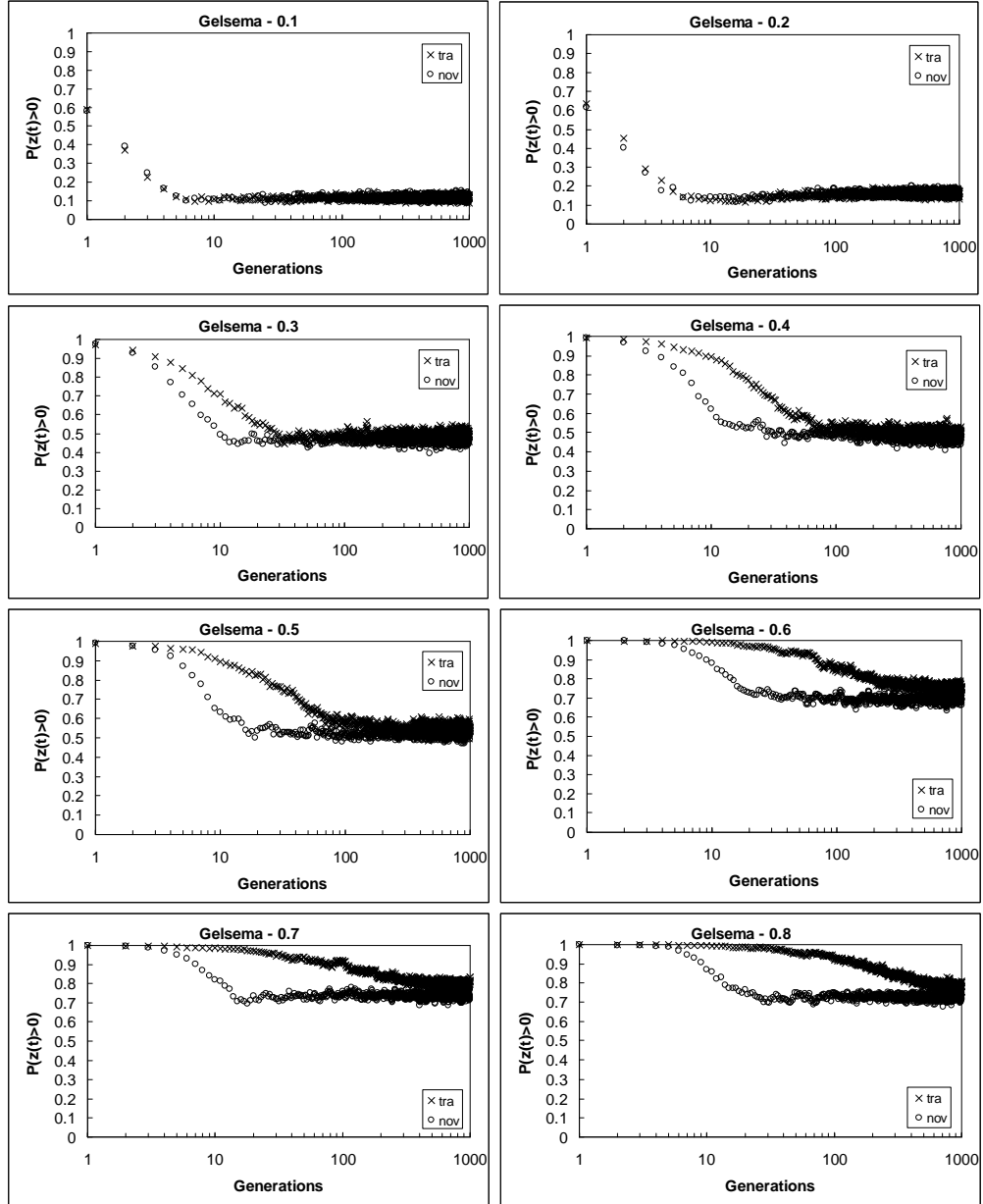


Figure 7: $P(z(t) > 0)$ results for eight Gelsema networks and $\rho \in \{0.1, \dots, 0.8\}$ under the traditional (tra) and the novel (nov) approaches.

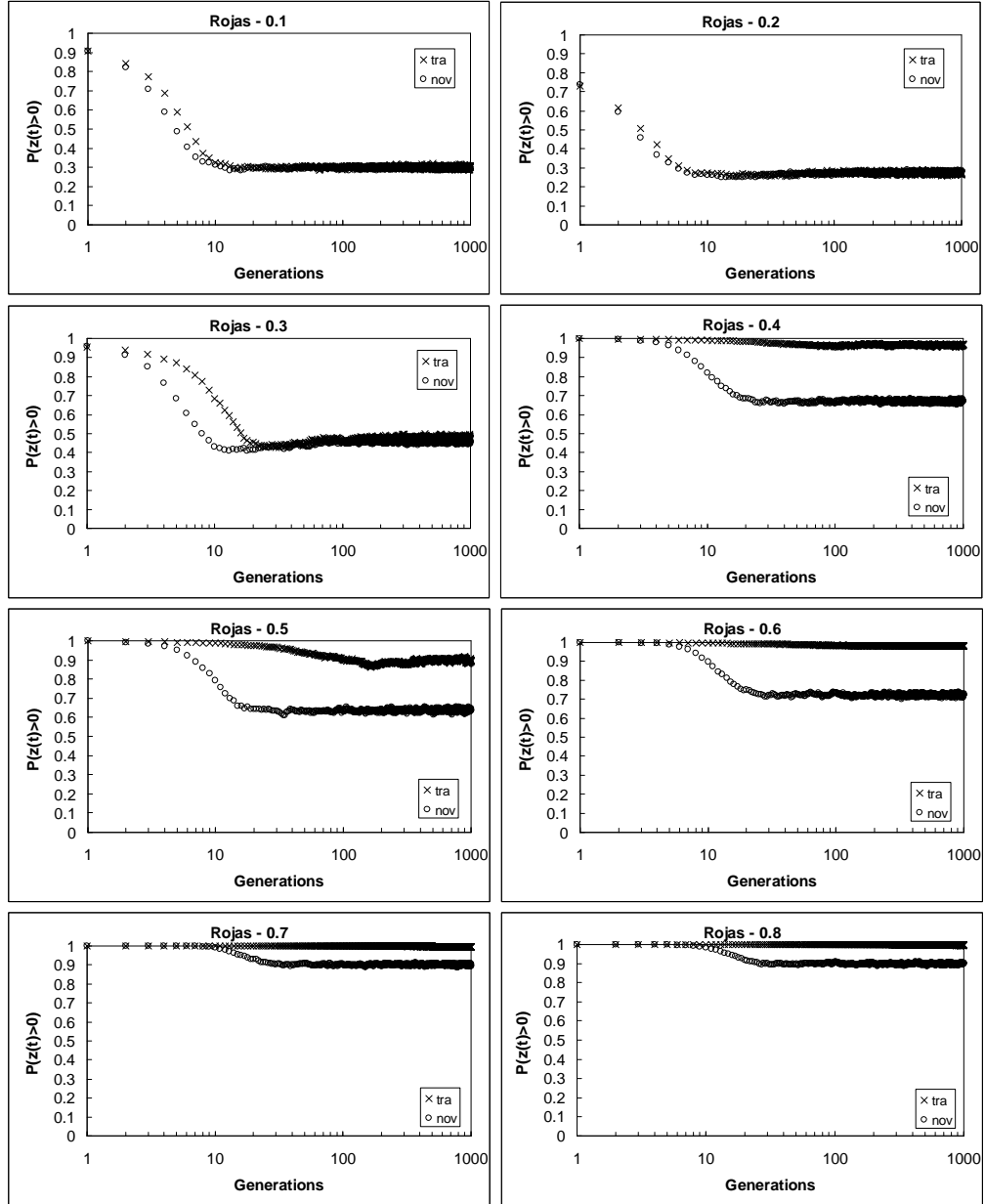


Figure 8: $P(z(t) > 0)$ results for eight Rojas networks and $\rho \in \{0.1, \dots, 0.8\}$ under the traditional (tra) and the novel (nov) approaches.

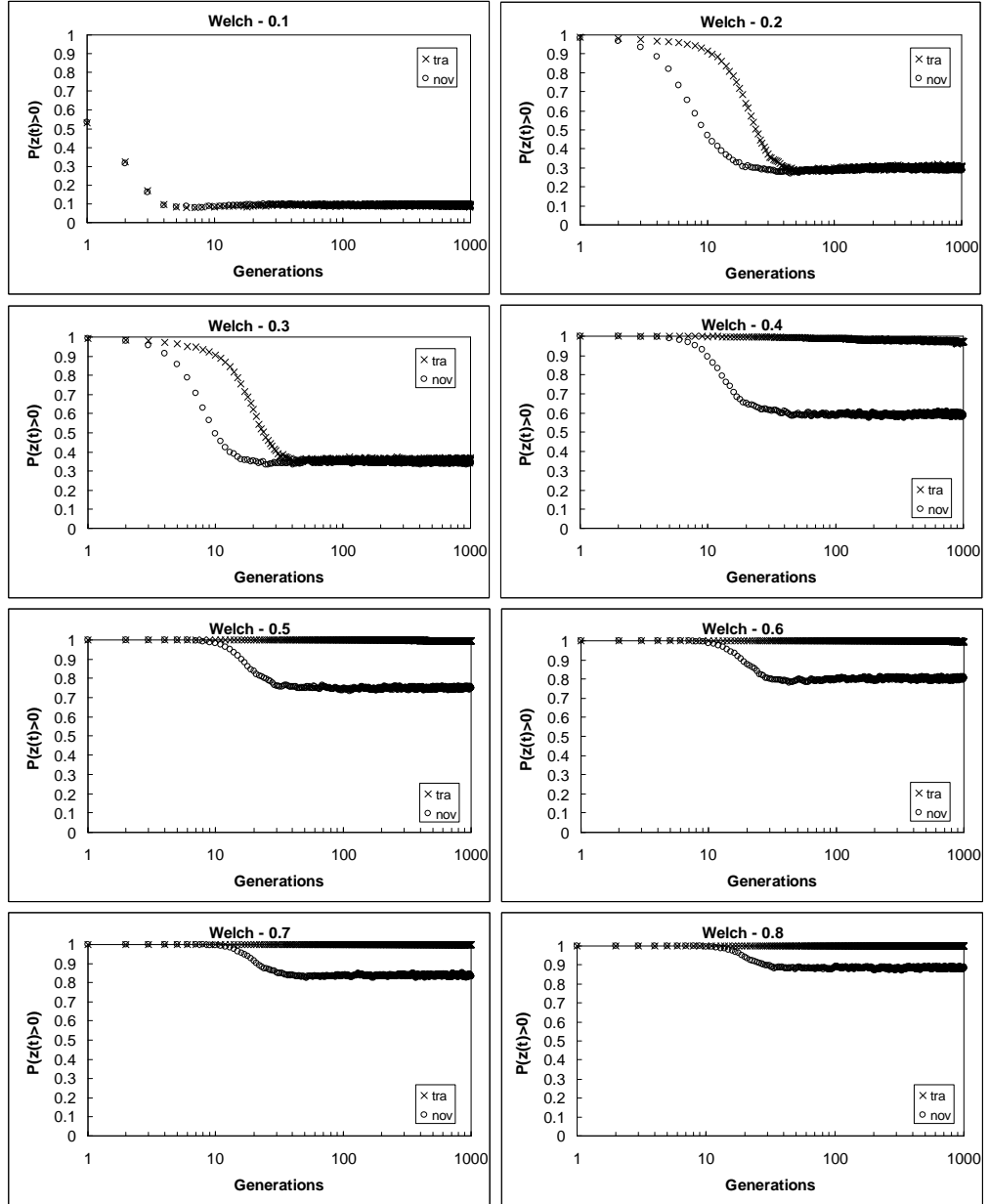


Figure 9: $P(z(t) > 0)$ results for eight Welch networks and $\rho \in \{0.1, \dots, 0.8\}$ under the traditional (tra) and the novel (nov) approaches.

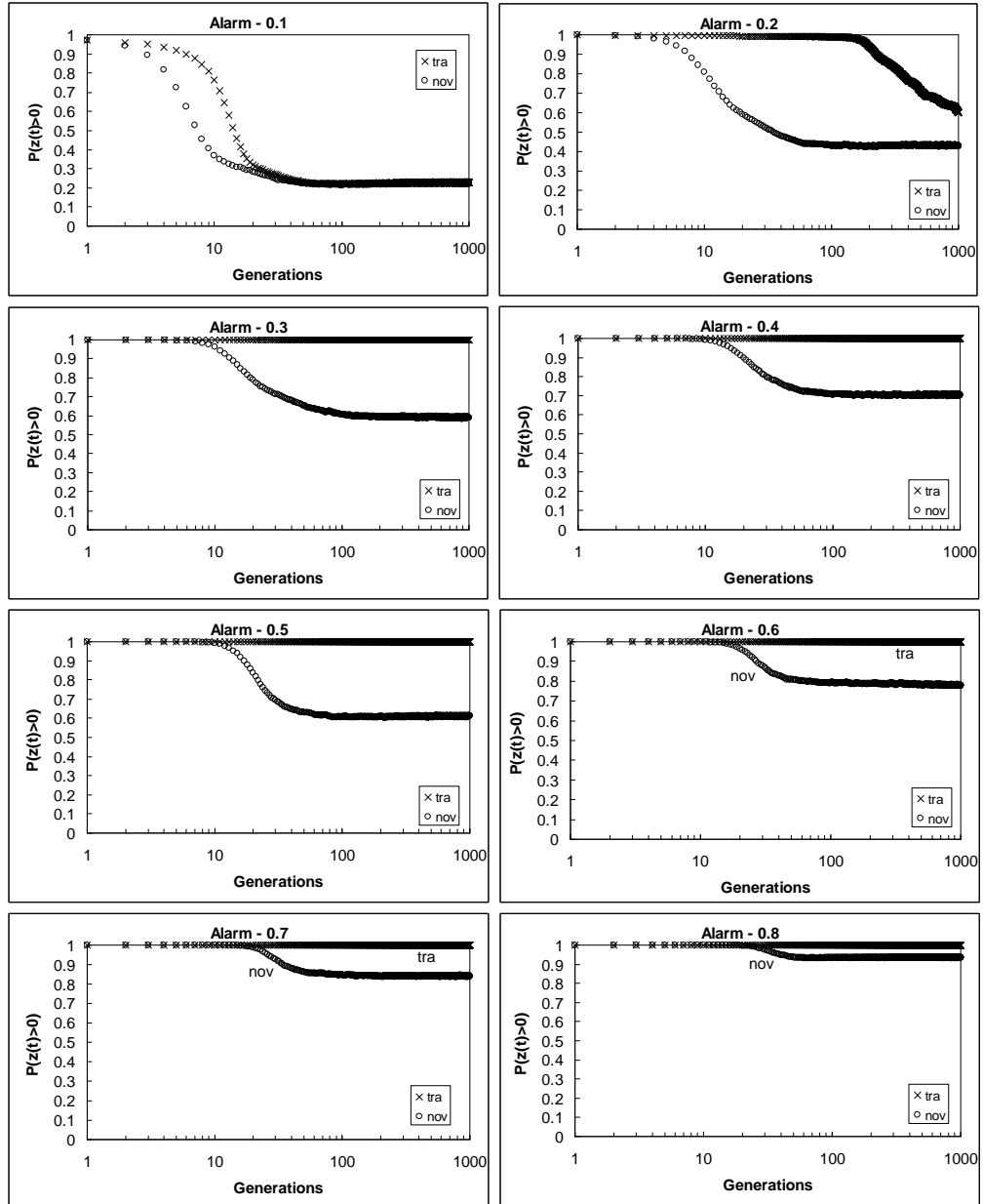


Figure 10: $P(z(t) > 0)$ results for eight Alarm networks and $\rho \in \{0.1, \dots, 0.8\}$ under the traditional (tra) and the novel (nov) approaches.

in Tables 2-5 are similar under both approaches for low ρ values; however, the differences both in best fitness and in accuracy for the two approaches are increasingly larger with ρ . The first piece of data in Tables 2-5 reporting outperformance by the novel approach can be observed between $\rho = 0.1$ and $\rho = 0.3$, depending on the particular type of network. It should be noted that even if the accuracy column in Table 5 for Alarm networks does not report significant differences (except for $\rho = 0.8$), the best fitness column in this table clearly demonstrates that these differences exist and that they are significant. In fact, performance differences between these approaches seem to get more pronounced as BNs with larger domain sizes are considered.

6.2 Performance Generation by Generation

Results obtained generation by generation in Figures 3-6 show that the novel approach performs increasingly better than the traditional approach as ρ grows. While the best-fitness convergence pattern for the novel approach remains almost the same for different ρ values, this pattern for the traditional approach experiences significant changes with ρ ; the greater the domain size of the considered BN is (see Table 1), the more significant the changes are. BN domain size also has an influence on the lowest ρ value that permits non-trivial differences between the two approaches to be observed: $\rho = 0.2$ for Alarm networks, $\rho = 0.2$ for Welch networks, $\rho = 0.3$ for Rojas networks, and $\rho = 0.3$ for Gelsema networks. In some of the graphs in Figures 3-6, namely Rojas 0.7-0.8, Welch 0.5-0.8, and Alarm 0.3-0.8, while the novel approach consistently converges to a good feasible solution by generation 100, the traditional approach can only supply explanations whose fitness is near zero even after 1000 generations. In this latter case, the traditional approach is conducting most of its search process in a zero-fitness plateau which, however, is efficiently avoided by the novel approach. Consequently, for BNs with large domain sizes—the ones for which an approximate inference method makes sense—the novel approach provides important inferential advantages, even for low ρ values such as 0.2 or 0.3.

The graphs for $P(z(t) > 0)$ in Figures 7-10 demonstrate the same global result as obtained for best fitness in Figures 3-6: The traditional approach is highly sensitive to ρ values, unlike the novel approach, which exhibits a more robust behavior. As ρ grows, the convergence time of $P(z(t) > 0)$ under the traditional approach increases significantly to the point that, for high ρ values (Rojas 0.6-0.8, Welch 0.4-0.8, and Alarm 0.3-0.8), only solutions violating one or more constraints exist in the population throughout the 1000 generations considered. It is interesting to observe the asymptotic behavior of $P(z(t) > 0)$: As ρ increases, $P(z(t) > 0)$ approaches one for high t values; this is due to the fact that, in such circumstances, CPTs contain an increasing number of zeros and, as a consequence, it becomes highly probable that crossover and mutation generate individuals violating at least one constraint.

The experimental results obtained for $P(z(t) > 0)$ support the validity of Theorem 2. By considering $t = 1$ (initial population), Table 7 shows $P(z(1) > 0)$ experimental results for the four type of networks under the traditional and the novel approaches. These values can be compared in Table 7 with the theoretical values given by Theorem 2. In general, the experimental results present a similar pattern to that of the theoretical values.

As discussed in Section 4.2, random walk represents a clear disadvantage for the traditional approach compared to the novel approach. The mean generation at which random walk ends under the traditional approach was measured over a set of runs for the experiments reported in Section 5.2. Table 8 contains these mean generation values

<i>Network</i>	$P(z(1) > 0)$		
	<i>Experimental</i> <i>Traditional</i>	<i>Novel</i>	<i>Theoretical</i> <i>Both Approaches</i>
Gelsema-0.1	0.588	0.581	0.537
Gelsema-0.2	0.632	0.618	0.794
Gelsema-0.3	0.97	0.975	0.913
Gelsema-0.4	0.991	0.99	0.965
Gelsema-0.5	0.989	0.991	0.987
Gelsema-0.6	0.999	1	0.995
Gelsema-0.7	0.998	0.999	0.998
Gelsema-0.8	0.999	0.999	0.999
Rojas-0.1	0.904	0.906	0.681
Rojas-0.2	0.729	0.738	0.905
Rojas-0.3	0.956	0.957	0.974
Rojas-0.4	0.999	0.999	0.993
Rojas-0.5	0.998	0.998	0.998
Rojas-0.6	0.999	0.999	1
Rojas-0.7	1	1	1
Rojas-0.8	1	1	1
Welch-0.1	0.531	0.534	0.806
Welch-0.2	0.987	0.985	0.966
Welch-0.3	0.99	0.991	0.994
Welch-0.4	1	1	0.999
Welch-0.5	1	1	1
Welch-0.6	1	1	1
Welch-0.7	1	1	1
Welch-0.8	1	1	1
Alarm-0.1	0.973	0.973	0.916
Alarm-0.2	0.998	0.998	0.994
Alarm-0.3	1	1	1
Alarm-0.4	1	1	1
Alarm-0.5	1	1	1
Alarm-0.6	1	1	1
Alarm-0.7	1	1	1
Alarm-0.8	1	1	1

Table 7: Experimental and theoretical values for $P(z(1) > 0)$.

for the subset of networks in which each of the 100 runs led to random-walk end by generation 1000. The table also includes theoretical values calculated from Theorem 3. Even if experimental and theoretical values are comparable, the correspondence is not exact for two main reasons. Firstly, there is a high variability regarding random-walk end among different runs. Secondly, random walk was approximated as the random generation of a population generation by generation, not taking into account the effects of parent selection, crossover, and mutation.

The number of variables is not a reliable indicator of hardness of a BN. If the BN is sparse, for example if it has a tree structure, then a structure-based algorithm can solve it in milliseconds. On the other hand, if the BN is not sparse, then the BN is

Network	Mean generation at which random walk ends	
	Experimental values	Theoretical values
Gelsema-0.1	1.01	1
Gelsema-0.2	1.01	1.11
Gelsema-0.3	7.43	1.67
Gelsema-0.4	20.65	3.32
Gelsema-0.5	29.67	7.94
Gelsema-0.6	123.17	21.51
Gelsema-0.7	122.17	64.47
Rojas-0.1	1	1
Rojas-0.2	1	1
Rojas-0.3	1.02	1.06
Rojas-0.4	10.97	1.9
Rojas-0.5	9.88	6.19
Rojas-0.6	36.18	27.59
Welch-0.1	1	1
Welch-0.2	1.3	1
Welch-0.3	1.38	1.77
Welch-0.4	45.63	8.92
Alarm-0.1	1	1
Alarm-0.2	1.3	1
Alarm-0.3	27.34	2.72

Table 8: Experimental and theoretical values for the mean generation at which random walk ends under the traditional approach. The following population sizes were used: $m = 10$ for Gelsema networks, $m = 100$ for Rojas networks, $m = 150$ for Welch networks, and $m = 1000$ for Alarm networks.

often beyond the scope of present-day algorithms and hardware. It is this latter second type of BNs we are ultimately interested in solving; this article enables scalability to larger BN. We believe that further improvements in scalability would require the use of additional and quite different techniques such as niching (Mengshoel and Goldberg, 1999; Mengshoel and Wilkins, 1998), stochastic local search (Hoos and Stützle, 2005), initialization (Mengshoel et al., 2007), or related methods, all beyond the scope of this paper.

7 Conclusion

This work develops and analyzes an approach to BN constraint handling in evolutionary algorithms that use tournament selection. The abductive inference approach is hybrid in that it combines feasibility considerations with the use of a penalty function. If both explanations (individuals) are infeasible, the individual with the smallest penalty function value, which measures the number of violated BN constraints, is the winner. In all other cases, the explanation with the greatest probability is the winner.

Approximate abductive inference in BNs through evolutionary methods has traditionally been based on assigning each candidate explanation a fitness equal to its joint probability, calculated from Equation 1. The present work demonstrates a limitation of the traditional approach as the number of zeros in CPTs grows. Frequently, this approach becomes almost useless because of the difficulty of finding explanations with

joint probability greater than zero, giving rise to a random walk that dramatically delays convergence to an optimal or approximately optimal solution.

To address this problem, this work has presented a novel tournament selection approach to approximate abductive inference in BNs. The underlying idea of the novel approach is the redefinition of the fitness function by taking into account, for each explanation, the number of family configurations leading to a conditional probability equal to zero. (A family configuration is an assignment of values to a node and its parents in the network.) In this way, the novel approach allows approximate abductive inference to be tackled as a constraint optimization problem, since a family configuration leading to a conditional probability equal to zero is said to originate a constraint violation. If all explanations in a tournament have zero probability, i.e., they are all infeasible, the explanation that violates fewer constraints wins the tournament.

The novel approach has the advantage that random walks are avoided in the search process. Besides, even in the case that no random walk would take place under the traditional approach, the novel approach might perform a more efficient search due to its ability to get rid of explanations violating constraints. This is an important capability for BNs with a number of zeros in their CPTs, since in such cases crossover and mutation generate individuals violating one or more constraints with high probability. Finally, when no zeros are present in the BN CPTs, both approaches are equivalent, i.e., just the joint conditional probability is relevant for the fitness function definition under both approaches.

The present work opens several research directions for the future. A blind mutation was here used such that any non-evidential gene could be mutated; it is possible that better performance could be obtained if just non-evidential genes giving rise to a constraint violation are mutated, i.e., the novel approach could benefit from the introduction of an *intelligent mutation*. Further advantages could perhaps be obtained from applying diversity maintenance techniques like niching (Mengshoel and Wilkins, 1998) or population initialization methods based on stochastic simulation algorithms for BNs (Cheng and Druzdzel, 2000). Adopting a more general point of view, this work has an impact on any search method that could be applied to approximate abductive inference in BNs; techniques like *simulated annealing* or *ant colony optimization* (Guo et al., 2004), among others, could greatly benefit from adopting the underlying idea of this work.

Acknowledgements

Ole J. Mengshoel gratefully acknowledges support by NASA under award NCC2-1426.

References

- Abdelbar, A. M. and Hedetniemi, S. M. (1998). Approximating MAPs for belief networks is NP-hard and other theorems. *Artificial Intelligence*, 102:21–38.
- Bäck, T., Fogel, D. B., and Michalewicz, Z., editors (2000a). *Evolutionary Computation 1: Basic Algorithms and Operators*. Institute of Physics Publishing, Bristol, UK.
- Bäck, T., Fogel, D. B., and Michalewicz, Z., editors (2000b). *Evolutionary Computation 2: Advanced Algorithms and Operators*. Institute of Physics Publishing, Bristol, UK.
- Beinlich, I. A., Suermondt, H. J., Chavez, R. M., and Cooper, G. F. (1989). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine (AIME'89)*, pages 246–256.
- Castillo, E., Gutiérrez, J. M., and Hadi, A. S. (1997). *Expert Systems and Probabilistic Network Models*. Springer-Verlag, New York.

- Cheng, J. and Druzdzel, M. J. (2000). AIS-BN: an adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research*, 13:155–188.
- Coello, C. A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287.
- Coello, C. A. and Montes, E. M. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16(3):193–203.
- Coello, C. A., van Veldhuizen, D. A., and Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, Boston, MA.
- Dawid, A. P. (1992). Applications of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, 2:25–36.
- de Campos, L. M., Gámez, J. A., and Moral, S. (2002). Partial abductive inference in Bayesian belief networks: an evolutionary computation approach by using problem-specific genetic operators. *IEEE Transactions on Evolutionary Computation*, 6:105–131.
- Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186:311–338.
- Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, Chichester, UK.
- Druzdzel, M. J. (1994). Some properties of joint probability distributions. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence (UAI'94)*, pages 187–194, Seattle, WA. Morgan Kaufmann, San Francisco, CA.
- Eiben, A. E. (2001). Evolutionary algorithms and constraint satisfaction: definitions, survey, methodology, and research directions. In Kallel, L., Naudts, B., and Rogers, A., editors, *Theoretical Aspects of Evolutionary Computing*, pages 13–58. Springer, Berlin, Heidelberg, New York.
- Fung, R. and del Favero, B. (1995). Applying Bayesian networks to information retrieval. *Communications of the ACM*, 38(3):42–57.
- Galán, S. F., Arroyo-Figueroa, G., Díez, F. J., and Sucar, L. E. (2007). Comparison of two types of event Bayesian networks: a case study. *Applied Artificial Intelligence*, 21(3):185–209.
- Gámez, J. A. (2004). Abductive inference in Bayesian networks: a review. In Gámez, J. A., Moral, S., and Salmerón, A., editors, *Advances in Bayesian networks*, pages 101–117. Springer, Berlin.
- Gelsema, E. S. (1995). Abductive reasoning in Bayesian belief networks using a genetic algorithm. *Pattern Recognition Letters*, 16:865–871.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.
- Goldberg, D. E. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*, 1:69–93.
- Guo, H., Boddhireddy, P. R., and Hsu, W. H. (2004). An ACO algorithm for the most probable explanation problem. In *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence (AI'2004)*, pages 778–790, Cairns, Australia. Springer, Berlin, Heidelberg.
- Heckerman, D. E., Horvitz, E. J., and Nathwani, B. N. (1992). Toward normative expert systems: Part I — The Pathfinder Project. *Methods of Information in Medicine*, 31:90–105.
- Heckerman, D. E. and Nathwani, B. N. (1992). Toward normative expert systems: Part II — Probability-based representations for efficient knowledge acquisition and inference. *Methods of Information in Medicine*, 31:106–116.

- Heckerman, D. E., Wellman, M., and Mandani, A. (1995). Real-world applications of Bayesian networks. *Communications of the ACM*, 38(3):24–26.
- Holland, J. H. (1985). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI. Second edition, The MIT Press, Boston, MA, 1992.
- Hoos, H. H. and Stützle, T. (2005). *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, San Francisco.
- Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998). The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'1998)*, pages 256–265, Madison, WI. Morgan Kaufmann, San Francisco, CA.
- Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Springer, Berlin.
- Jiménez, F. and Verdegay, J. L. (1999). Evolutionary techniques for constrained optimization problems. In *Proceedings of the 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, Aachen, Germany. Verlag Mainz Publishers, Aachen, Germany.
- Larrañaga, P. and Lozano, J. A., editors (2002). *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston, Dordrecht, London.
- Lin, R., Galper, A., and Shachter, R. (1990). Abductive inference using probabilistic networks: randomized search techniques. Technical Report KSL-90-73, Knowledge Systems Laboratory, Stanford University, CA.
- Mengshoel, O. J. and Goldberg, D. E. (1999). Probabilistic crowding: deterministic crowding with probabilistic replacement. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, pages 409–416, Orlando, FL.
- Mengshoel, O. J., Roth, D., and Wilkins, D. C. (2007). Initialization and restart in stochastic local search: computing a most probable explanation in Bayesian networks. *Submitted for publication*.
- Mengshoel, O. J. and Wilkins, D. C. (1998). Genetic algorithms for belief network inference: the role of scaling and niching. In *Proceedings of the 7th International Conference on Evolutionary Programming*, pages 547–556, San Diego, CA. Springer-Verlag, London, UK.
- Mengshoel, O. J., Wilkins, D. C., and Roth, D. (2006). Controlled generation of hard and easy Bayesian networks: impact on maximal clique size in tree clustering. *Artificial Intelligence*, 170(16-17):1137–1174.
- Michalewicz, Z. and Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimisation problems. *Evolutionary Computation*, 4(1):1–32.
- Nilsson, D. (1998). An efficient algorithm for finding the M most probable configurations in Bayesian networks. *Statistics and Computing*, 9:159–173.
- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, CA. Revised second printing, 1991.
- Pennock, D. M., Horvitz, E., Lawrence, S., and Giles, C. L. (2000). Collaborative filtering by personality diagnosis: a hybrid memory- and model-based approach. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI'2000)*, pages 473–480, Stanford, CA. Morgan Kaufmann, San Francisco, CA.
- Rimey, R. D. and Brown, C. M. (1994). Control of selective perception using Bayes nets and decision theory. *International Journal of Computer Vision*, 12(2/3):173–207.
- Rojas-Guzmán, C. and Kramer, M. A. (1993). Galgo: a genetic algorithm decision support tool for complex uncertain systems modeled with Bayesian belief networks. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence (UAI'93)*, pages 368–375, Washington, DC. Morgan Kaufmann, San Francisco, CA.

- Rojas-Guzmán, C. and Kramer, M. A. (1996). An evolutionary computing approach to probabilistic reasoning in Bayesian networks. *Evolutionary Computation*, 4:57–85.
- Selman, B. and Kautz, H. (1993). Domain-independent extensions to GSAT: solving large structured satisfiability problems. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 290–295, Chambéry, France.
- Seroussi, B. and Goldmard, J. L. (1994). An algorithm directly finding the K most probable configurations in Bayesian networks. *International Journal of Approximate Reasoning*, 11:205–233.
- Shimony, E. (1994). Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, 68:399–410.
- Thierens, D. and Goldberg, D. E. (1994). Convergence models of genetic algorithm selection schemes. In *Proceedings of 3rd Conference on Parallel Problem Solving from Nature*, pages 119–129, Jerusalem, Israel. Springer-Verlag, Berlin.
- Welch, R. L. (1996). Real-time estimation of Bayesian networks. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI'96)*, pages 533–544, Portland, OR. Morgan Kaufmann, San Francisco, CA.
- Zhong, X. and Santos Jr, E. (2000). Directing genetic algorithms for probabilistic reasoning through reinforcement learning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 8:167–186.